IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

1. (original) A method of performing a query on a Markup document, which comprises:

receiving a query;

designing a plurality of filters to reflect a structural linkage of a condition tree representing the query, wherein designing the plurality of filters comprises:

designing a highest-level filter that can become active only if an event-based parser indicates that an element for which the highest-level filter is searching has been found, and

designing a lowest-level filter that can become active only when the highest-level filter has become active and when the parser indicates that an element for which the lowest-level filter is searching has been parsed;

parsing the Markup document; and

checking the lowest-level filter to determine whether it has found the element for which it has been searching.

2. (original) The method according to claim 1, wherein the step of designing the plurality of filters further comprises:

designing at least one intermediate-level filter that can become active only when the highest-level filter has become active and when the parser indicates that an element for which the intermediate-level filter is searching has been parsed; and

designing the lowest-level filter to become active only when the intermediate-level filter has become active.

3. (original) The method according to claim 1, further comprising:

defining the lowest-level filter as a first lowest-level filter;

designing a second lowest-level filter that can become active only when the highest-level filter has become active and when the parser indicates that an element for which the lowest-level filter is searching has been parsed; and

checking the second lowest-level filter to determine whether it has found the element for which it has been searching.

4. (original) The method according to claim 3, further comprising:

designing a value filter that will become active only when the highest-level filter has become active and when the parser indicates that an element for which the value filter is searching has been parsed; and

if the first lowest-level filter has found the element for which it has been searching and the second lowest-level filter has found the element for which it has been searching, obtaining an element from the value filter that is linked to the elements in the first lowest-level filter and in the second lowest-level filter.

5. (original) The method according to claim 1, further comprising:

designing a value filter that will become active only when the highest-level filter has become active and when the parser indicates that an element for which the value filter is searching has been parsed; and

if the lowest-level filter has found the element for which it has been searching, obtaining an element from the value filter that is linked to the element in the lowest-level filter.

6. (original) A computer-readable medium storing a program for controlling a computer to perform a method of performing a query on a Markup document, which comprises:

receiving a query;

designing a plurality of filters to reflect a structural linkage of a condition tree representing the query, wherein designing the plurality of filters comprises:

designing a highest-level filter that can become active only if an event-based parser indicates that an element for which the highest-level filter is searching has been found, and

designing a lowest-level filter that can become active only when the highest-level filter has become active and when the parser indicates that an element for which the lowest-level filter is searching has been parsed;

parsing the Markup document; and

checking the lowest-level filter to determine whether it has found the element for which it has been searching.

7. (original) A device to perform a query on a Markup document, comprising:

a receiver to receive a query;

a design unit to design a plurality of filters to reflect a structural linkage of a condition tree representing the query, the design unit comprising:

a highest level designer to design a highest-level filter that can become active only if an event-based parser indicates that an element for which the highest-level filter is searching has been found, and

a lowest-level designer to design a filter that can become active only when the highest-level filter has become active and when the parser indicates that an element for which the lowest-level filter is searching has been parsed;

a parser to parse the Markup document; and

a checker to check the lowest-level filter to determine whether it has found the element for which it has been searching.